UNCLASSIFIED

NONLINEAR ADAPTIVE CONTROL OF AGILE ANTI-AIR MISSILES USING NEURAL NETWORKS

Michael B. McFarland[*] and Anthony J. Calise[**]
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

Abstract

Research has shown that neural networks can be used to improve upon approximate dynamic inversion controllers in the case of uncertain nonlinear systems. In one possible architecture, the neural network adaptively cancels linearization errors through on-line learning. Learning may be accomplished by a simple weight update rule derived from Lyapunov theory, thus assuring the stability of the closed-loop system. In this paper, the authors discuss the evolution of this methodology and its application in a bank-to-turn autopilot design for an agile anti-air missile. Additional consideration is given to robustness of the proposed controller. First, a control scheme based on approximate inversion of the vehicle dynamics is presented. This nonlinear control system is then augmented by the addition of a feedforward neural network with on-line learning. Finally, the resulting control law is demonstrated in a nonlinear simulation and its performance is evaluated relative to a more traditional gain-scheduled linear autopilot.

Introduction

Advances in fighter aircraft technology continue to create new challenges for designers of anti-air weapon systems. The introduction of low-observable aircraft has increased the need for small, lightweight missiles. This, in turn, has led to various control problems associated with air-breathing propulsion, asymmetric airframes, and greatly reduced aerodynamic control surface area. Similarly, the advent of supermaneuverable aircraft has motivated efforts aimed at expanding the missile's flight envelope to include high angle of attack conditions. Accordingly, some proposed next-generation missiles employ nonaerodynamic, propulsive control effectors similar to those associated with supermaneuverable aircraft. These so-called "agile" missiles possess enhanced range as a result of their ability to execute rapid propulsive heading changes during the boost phase. Furthermore, these weapons can be deployed during high angle of attack maneuvers, and may even engage targets in the rear hemisphere relative to the launch aircraft.

The dynamics of an agile missile flying in bank-to-turn mode at a high angle of attack are inherently nonlinear and may vary rapidly with time. Furthermore, these dynamics are highly uncertain since aerodynamic data for vehicles operating under such conditions is difficult to obtain and may in fact be a poor approximation to the actual dynamics. These and other concerns have prompted researchers to look beyond the classical methods which have historically dominated the field of missile autopilot design and to robust, nonlinear, and "intelligent" control.

Most nonlinear control techniques are based on linearizing the equations of motion by the application of nonlinear feedback. Known variously as feedback linearization or dynamic inversion, this method relies heavily on knowledge of the plant dynamics. An early application of this theory to the missile autopilot design problem is found in Ref. 1, while Ref. 2 presents a more sophisticated approach involving variable structure control. More recently, neural networks have emerged as a means of explicitly accounting for uncertainties in the plant dynamics. Their on-line learning and functional approximation capabilities make neural networks an excellent candidate for this application. Ref. 3 is one example in which neural networks are investigated for nonlinear control of missiles.

This paper concerns a neural network based approach to direct adaptive control of nonlinear systems which originated in Ref. 4 and was further developed in Ref. 5. In the proposed architecture, a simple dynamic inversion controller approximately linearizes the vehicle dynamics. This controller is augmented by a neural network which acts to improve the linearization by adaptively canceling inversion errors in real-time. The neural network implementation features a stable on-line learning algorithm derived from Lyapunov theory. While Ref's. 4 and 5 dealt specifically with fighter aircraft and helicopter applications, respectively, the work described in this paper adapts these previous efforts for use in control of agile missiles. The result is an autopilot which combines the best features of dynamic inversion, neural networks, and adaptive control to potentially increase the effectiveness and versatility of tomorrow's missile systems.

1

UNCLASSIFIED

The preliminary study presented in Ref. 6 revealed that neural networks are indeed capable of attaining sufficiently high learning rates to make adaptation feasible during even the most demanding aerial engagements. In fact, previous work documented in Ref. 7 shows that the methodology considered here compares favorably to traditional gain-scheduled linear methods for this application. This paper summarizes the results of earlier research with additional comments on robustness.

The paper begins with a review of the proposed controller architecture and its evolution to date. The methodology in question is then applied to an agile anti-air missile autopilot design problem. First, the baseline control scheme consisting of an approximate inversion of the missile's six-degree-of-freedom nonlinear dynamics is presented. Neural networks are then designed to enhance this nonlinear controller. Simulation results comparing this technique to a traditional gain-scheduled implementation are presented, and the effects of neural network topology are investigated. Finally, conclusions and future research directions are discussed.

Control Design Methodology

Consider a block-triangular nonlinear system with the following structure

$$\dot{x}_1 = f_1(x_1) + g_1(x_1)x_2$$
$$\dot{x}_2 = f_2(x_1, x_2, u) \quad (1)$$

where $x_1, x_2, u \in \Re^n$ and $g_1(x_1)$ remains nonsingular for all x_1. For reference, a block diagram of this system is included in Fig. 1.
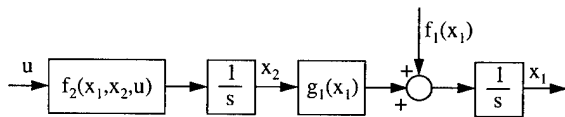

Figure 1: Open-Loop Nonlinear System

Approximate Nonlinear Control

Suppose that the dynamics of the x_1 subsystem are well-known, but the nonlinearity f_2 is only known approximately. Furthermore, suppose a stabilizing controller $\alpha_1(x_1, t)$ for the x_1 subsystem with x_2 as input is also known. One example is the dynamic inversion controller described by

$$\alpha_1(x_1, t) = g_1^{-1}(x_1)\left[K_1(x_{1c} - x_1) + \dot{x}_{1c}(t) - f_1(x_1)\right] \quad (2)$$

where x_{1c} denotes the command vector. Note, however, that $\alpha_1(x_1, t)$ need not be an inverting controller. If, for instance, the plant dynamics include so-called

"beneficial" nonlinearities, these terms need not be canceled by the nonlinear feedback. This is "lean" nonlinear control as described in Ref. 8 with regard to integrator backstepping. Defining the desired error variables,

$$\tilde{x}_1 = x_{1c}(t) - x_1$$
$$\tilde{x}_2 = \alpha_1(x_1, t) - x_2 \quad (3)$$

we then close the loop on the x_1 subsystem, writing

$$\dot{\tilde{x}}_1 = -K_1\tilde{x}_1 + g_1(x_1)\tilde{x}_2$$
$$\dot{x}_2 = f_2(x_1, x_2, u) \quad (4)$$

A technique derived from integrator backstepping will now be used to construct a stabilizing controller for Eq. (1). We first turn our attention to the x_2 subsystem. In Eq. (4), the x_2 subsystem may be rewritten as

$$\dot{x}_2 = v$$
$$v = f_2(x_1, x_2, u) \quad (5)$$

where $v(t) \in R^n$ is a pseudo-control input. Provided that the mapping $f_2(x_1, x_2, u)$ is invertible and full-state feedback is available, the definition of v above provides a linearizing transformation of the control. The inverse transformation is expressed by

$$u = f_2^{-1}(x_1, x_2, v) \quad (6)$$

and must be computed in real-time when the control is implemented. If the mapping f_2 is known and its inverse is computed accurately, then the system is exactly linearized. Since f_2 is uncertain, exact linearization is impossible and dynamic inversion results in:

$$\dot{x}_2 = v + \Delta(x_1, x_2, v) \quad (7)$$

where

$$\Delta(x_1, x_2, v) = f_2(x_1, x_2, \hat{u}) - \hat{f}_2(x_1, x_2, \hat{u}) \quad (8)$$

and $\Delta: R^n \times R^n \times R^n \to R^n$ is a nonlinear mapping representing the inversion error. In the above equation, $\hat{u} = \hat{f}_2^{-1}(x_1, x_2, v)$ describes the approximate inverse mapping. Fig. 2 below illustrates this approximate dynamic inversion process.
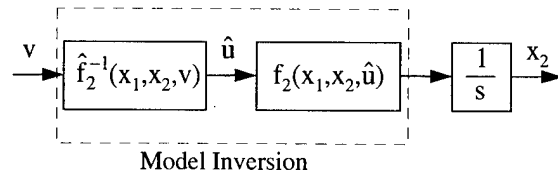

Model Inversion

Figure 2: Approximate Dynamic Inversion

In the ideal case, integrator backstepping may be used to construct an expression for $v(t)$ which stabilizes the errors $\tilde{x}_1$ and $\tilde{x}_2$. When there is non-zero inversion error, this expression is augmented by an adaptive term

2

which compensates for the error. This yields the following expression for the pseudo-control v(t):

$$v(t) = g_1^T(x_1)\tilde{x}_1 + \dot{\alpha}_1(x_1,t) + K_2\tilde{x}_2 - \hat{v}_{ad}(t) \quad (6)$$

where $\hat{v}_{ad}(t)$ is the adaptive control contribution to the pseudo-control. The first term in Eq. (6) is included to offset coupling between the $\tilde{x}_1$ and $\tilde{x}_2$ dynamics. It has become commonplace in applications of this type to reduce such coupling by enforcing time-scale separation between $\tilde{x}_1$ and $\tilde{x}_2$. In that case, the first term in Eq. (6) may be neglected. The second term is similar to the "command derivative" term in model-following control, and is necessary to achieve tracking of arbitrary continuous trajectories. For slowly varying commands, this term may also be neglected. Fig. 3 depicts the adaptive control architecture for the x_2 subsystem.
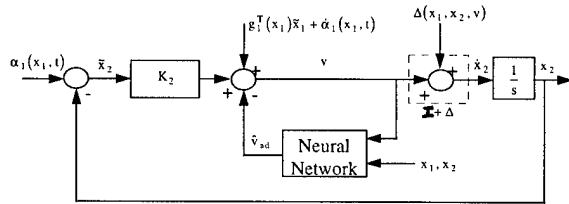

Figure 3: Adaptive Control Architecture

Neural Network Based Adaptation

In Ref. 4, the adaptive control is assigned the form

$$\hat{v}_{ad_i}(t) = \sum_{j=1}^{N} \hat{w}_{ij}(t)\beta_{ij}(x_1,x_2,v)$$
$$= \hat{w}_i^T(t)\beta_i(x_1,x_2,v) \quad (7)$$

for $i = 1,2,...,n$ with scalar weights $\hat{w}_{ij}(t)$ and the N-dimensional set of basis functions $\beta_{ij}(x,\dot{x},v)$. This approximation to the inversion error may be realized by any neural network which is linear in its adjustable parameters, often referred to as a "single-layer" network. One common architecture which satisfies this condition is the Sigma-Pi network depicted in Fig. 4.
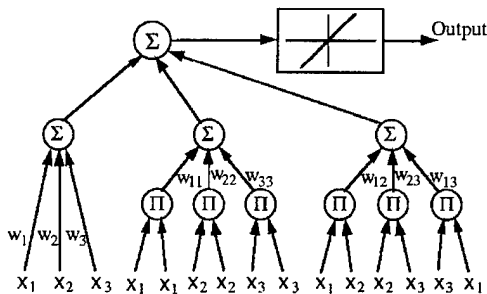

Figure 4: Sigma-Pi Neural Network

Using Lyapunov stability theory, it can be shown that the adaptive control achieves global asymptotic stability when the weight update rule is chosen as

$$\dot{\hat{w}}_i = \begin{cases} -\gamma_i\tilde{x}_{2_i}\beta_i(x_1,x_2,v) & \forall \left|\tilde{x}_{2_i}\right| > \varepsilon_i \\ 0 & \forall \left|\tilde{x}_{2_i}\right| \le \varepsilon_i \end{cases} \quad (8)$$
$$i = 1,2,...n$$

where $\gamma_i > 0$ is an adaptation gain, or learning rate, and ε_i denotes the magnitude of a deadzone introduced as an element of the stability proof as in Ref. 4

Note from Fig. 3 that the network input v depends on the current network output. Thus, a critical assumption in the stability proof involves the existence of a fixed-point solution for the output $\hat{v}_{ad}$. Such a condition is guaranteed when bounded basis functions β_i are used. Furthermore, when a stable fixed-point exists, a simple iterative scheme may be employed to compute the network output. A deadzone is required to account for the fact that the network is incapable of exactly representing the inversion error (Δ) by employing a finite set of basis functions.

Robustness Modifications

Throughout the development presented above, exact knowledge of the order of the open-loop plant is implicitly assumed. While the design is adaptive to uncertain nonlinearities, no allowance is made for unmodeled dynamics. Consider, for example, the open-loop system augmented by unmodeled dynamics $\Delta(s)$ at the input

$$\dot{x}_1 = f_1(x_1,t) + g_1(x_1,t)x_2$$
$$\dot{x}_2 = f_2(x,\overline{u},t) \quad (9)$$
$$\overline{u} = (1 + \mu\Delta(s))u$$

with $\mu \ge 0$. Although such a result is beyond the scope of this paper, it has been shown that under certain mild assumptions, the control law described above can be modified to achieve robust stability for sufficiently small values of the parameter μ. This is accomplished essentially by introducing additional dynamics into the nonlinear feedback used to compute u in Eq. 6. The reader is referred to Ref. 9 for details.

Missile Autopilot Application

In this section, the neural network-based adaptive control methodology described above is used to design a nonlinear autopilot for a next-generation anti-air missile depicted in Fig. 5. This vehicle is described in greater detail in Ref's. 10 and 11. In the current

scenario, the objective is to design a bank-to-turn autopilot which tracks external guidance commands in angle of attack and bank angle while holding sideslip near zero.
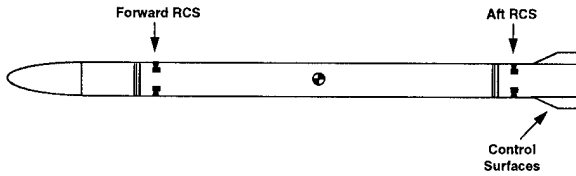

Figure 5: Agile Missile Configuration

Approximate Dynamic Inversion

The motion of a symmetric missile about its velocity vector may be described by the following equations.[12]

$$\dot{V} = \left(a_x \cos\alpha + a_z \sin\alpha\right)\cos\beta + a_y \sin\beta$$

$$\dot{\alpha} = q - \left(r\sin\alpha + p\cos\alpha\right)\tan\beta + \frac{-a_x \sin\alpha + a_z \cos\alpha}{V\cos\beta} \qquad (10)$$

$$\dot{\beta} = p\sin\alpha - r\cos\alpha - \frac{\left(a_x \cos\alpha + a_z \sin\alpha\right)\sin\beta - a_y \cos\beta}{V}$$

where V, α, and β denote airspeed, angle of attack, and sideslip angle. Here, a_x, a_y, and a_z are the body-axis components of acceleration (including gravitational effects) while p, q, and r are the body-axis angular rates. The moment equations have the form

$$\dot{p} = \frac{L}{I_{xx}}$$

$$\dot{q} = \frac{M}{I_{yy}} + \left(1 - \frac{I_{xx}}{I_{yy}}\right)pr \qquad (11)$$

$$\dot{r} = \frac{N}{I_{yy}} - \left(1 - \frac{I_{xx}}{I_{yy}}\right)pq$$

where L, M, and N represent aerodynamic moments about the body axes. Finally, I_{xx} and I_{yy} are rolling and pitching moments of inertia, respectively, and are assumed to have nearly constant values.

In order to simplify roll control in bank-to-turn flight, we now introduce an aerodynamic bank angle (μ) about the velocity vector. The bank angle dynamics are described by Eq. (12).

$$\dot{\mu} = \left(\frac{p\cos\alpha + r\sin\alpha}{\cos\beta}\right) + \frac{\left(a_x \sin\alpha - a_z \cos\alpha\right)\tan\beta}{V} \qquad (12)$$

The following development assumes the presence of a guidance law which commands angle of attack, sideslip angle, and bank angle. There are several possible

alternative choices for the roll command, including body roll angle (ϕ), body roll rate (p), and stability-axis roll rate ($p_s = p\cos\alpha + r\sin\alpha$).

Introducing the more compact notation,

$$x = \begin{Bmatrix} \alpha \\ \beta \\ \mu \end{Bmatrix} \qquad \omega = \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \qquad (13)$$

the last two of Eq's. (10) may be rewritten along with Eq. (12) as

$$\dot{x} = a_f + T(x)\omega \qquad (14)$$

where $T(x)$ and a_f are given by Eq. (15).

$$T(x) = \begin{bmatrix} -\cos\alpha\tan\beta & 1 & -\sin\alpha\tan\beta \\ \sin\alpha & 0 & -\cos\alpha \\ \cos\alpha\sec\beta & 0 & \sin\alpha\sec\beta \end{bmatrix}$$

$$a_f = \begin{Bmatrix} \dfrac{-a_x \sin\alpha + a_z \cos\alpha}{V\cos\beta} \\ \dfrac{-\left(a_x \cos\alpha + a_z \sin\alpha\right)\sin\beta + a_y \cos\beta}{V} \\ \dfrac{\left(a_x \sin\alpha - a_z \cos\alpha\right)\tan\beta}{V} \end{Bmatrix} \qquad (15)$$

The autopilot design methodology presented above will now be applied to the system described by Eq. (14) and Eq. (11). First, body-axis angular rate commands are computed from Eq. (2) as follows:

$$\overline{\omega}_c = T^{-1}(x)\left(K_x \tilde{x} + \dot{x}_c - a_f\right) \qquad (16)$$

The gain matrix K_x may be chosen as diagonal with positive elements. An integral term may also be included in Eq. (16), but has been omitted for simplicity. No adaptive control is necessary at this stage of the design, since the nonlinear dynamics are sufficiently well-known. Eq. (16) makes use of available acceleration and velocity information, as well as estimates of angle of attack and sideslip angle which are constructed from inertial data. The accelerometer measurements are filtered and biased appropriately to account for gravitational effects. Note that the matrix $T(x)$ may be inverted, since it is nonsingular for all α and β except $\beta=\pi/2$, which should not be encountered in bank-to-turn flight.

Next, a first-order filter is applied to the guidance commands ($\overline{\alpha}_c$, $\overline{\beta}_c$, and $\overline{\mu}_c$). Its outputs are the filtered commands and their rates.

$$x_c = \begin{Bmatrix} \alpha_c \\ \beta_c \\ \mu_c \end{Bmatrix} \qquad \dot{x}_c = \begin{Bmatrix} \dot{\alpha}_c \\ \dot{\beta}_c \\ \dot{\mu}_c \end{Bmatrix} \qquad (17)$$

At this point in the control design procedure, we turn our attention to the body-axis angular rates. First, a pseudo-control input, v, is defined by

$$\dot{\omega} = v$$
$$v = f\left(x, \omega, \hat{\delta}\right) \quad (18)$$

where the function $f(\cdot)$ refers to the right-hand-side of Eq. (11). The missile's body-axis angular rate dynamics must be approximately inverted to determine the required control input. For an approximate inversion, the body-axis moments are represented linearly:

$$L \approx \hat{L} = L_\beta \beta + L_p p + L_r r + L_{\delta p} \hat{\delta}_p + L_{\delta r} \hat{\delta}_r$$

$$M \approx \hat{M} = M_\alpha \alpha + M_q q + M_{\delta q} \hat{\delta}_q \quad (19)$$

$$N \approx \hat{N} = N_\beta \beta + N_p p + N_r r + N_{\delta p} \hat{\delta}_p + N_{\delta r} \hat{\delta}_r$$

The body angular rate dynamics of Eq. (11) are then rewritten as

$$\dot{\omega} = f\left(x, \omega, \hat{\delta}\right) = \hat{F}(x, \omega) + \hat{B}\begin{Bmatrix} \hat{\delta}_p \\ \hat{\delta}_q \\ \hat{\delta}_r \end{Bmatrix} + \Delta\left(x, \omega, \hat{\delta}\right) \quad (20)$$

where f is approximated linearly by introducing

$$\hat{F} = \begin{Bmatrix} \hat{F}_p \\ \hat{F}_q \\ \hat{F}_r \end{Bmatrix} \quad \text{and} \quad \hat{B} = \begin{bmatrix} \dfrac{L_{\delta p}}{I_{xx}} & 0 & \dfrac{L_{\delta r}}{I_{xx}} \\ 0 & \dfrac{M_{\delta q}}{I_{yy}} & 0 \\ \dfrac{-N_{\delta p}}{I_{xx}} & 0 & \dfrac{N_{\delta r}}{I_{yy}} \end{bmatrix} \quad (21)$$

with

$$\hat{F}_p = \frac{L_\beta \beta + L_p p + L_r r}{I_{xx}} \quad (22)$$

$$\hat{F}_q = \frac{M_\alpha \alpha + M_q q}{I_{yy}} + \left(1 - \frac{I_{xx}}{I_{yy}}\right) pr \quad (23)$$

$$\hat{F}_r = \frac{N_\beta \beta + N_p p + N_r r}{I_{yy}} + \left(1 - \frac{I_{xx}}{I_{yy}}\right) pq \quad (24)$$

Performing the approximate dynamic inversion, the control input ($\hat{\delta}$) is then given by Eq. (25).

$$\hat{\delta} = \hat{B}^{-1}\left(v - \hat{F}\right) \quad (25)$$

The next step in the control design is the computation of the pseudo-control input (v). Here, the commands generated by Eq. (16) are filtered to obtain

$$\omega_c = \begin{Bmatrix} p_c \\ q_c \\ r_c \end{Bmatrix} \quad \dot{\omega}_c = \begin{Bmatrix} \dot{p}_c \\ \dot{q}_c \\ \dot{r}_c \end{Bmatrix} \quad (26)$$

which are used in Eq. (6) to obtain

$$v = K_\omega \tilde{\omega} + \dot{\omega}_c - \hat{v}_{ad} \quad (27)$$

where $\hat{v}_{ad}$ is the adaptive control component which is included to cancel nonlinear inversion errors. As in the

outer loop, K_ω is a diagonal matrix of positive gains and the error variable $\tilde{\omega} = \omega_c - \omega$ has been introduced. Note that we have neglected the coupling term $+g_1^T(x_1)\tilde{x}_1$ from Eq. (6). We will instead choose K_x and K_ω to impose time-scale separation upon the closed-loop error dynamics.

In practice, the stability and control derivatives in Eq. (20) could be scheduled as a function of flight condition, along with trim values of the control inputs. This, however, presumes that an accurate full-envelope vehicle model is available. In the current implementation, the moment coefficients are treated as constants, shifting the burden of gain scheduling to the neural network. The dynamic inversion control law may be simplified further by introducing additional approximations as simulation results warrant. The neural network would then be required to additionally compensate for those effects which are neglected by the dynamic inversion.

In the actual implementation, the inputs computed by Eq. (25) are distributed among the missile's aerodynamic control surfaces and RCS thrusters using any suitable control allocation algorithm.

Neural Network Architecture

Substituting Eq. (16) into Eq. (14) and Eq. (25) into Eq. (20), the closed-loop error dynamics can be written as follows:

$$\dot{\tilde{x}} = -K_x \tilde{x}$$
$$\dot{\tilde{\omega}} = -K_\omega \tilde{\omega} - \hat{v}_{ad} + \Delta(x, \omega, v) \quad (28)$$

Any simple linearly parameterized feedforward neural network which is capable of approximately reconstructing the nonlinear inversion error (Δ) may be used to compute the adaptive contribution to the pseudo-control. The adaptive pseudo-control input in each channel is therefore described by Eq. (7) presented previously. Moreover, a stable learning rule is again given by Eq. (8).

Various neural network topologies and choices of inputs have been considered in this study. The results presented in the following section use a second-order Sigma-Pi network with inputs chosen as normalized values of x, ω, and v as well as a constant bias term.

Simulation Results

Results were obtained using the neural-adaptive nonlinear autopilot described above in a nonlinear, six-degree-of-freedom (6-DOF) simulation of a proposed agile anti-air missile. In this particular simulation, RCS thrusters and BTT guidance are only employed during
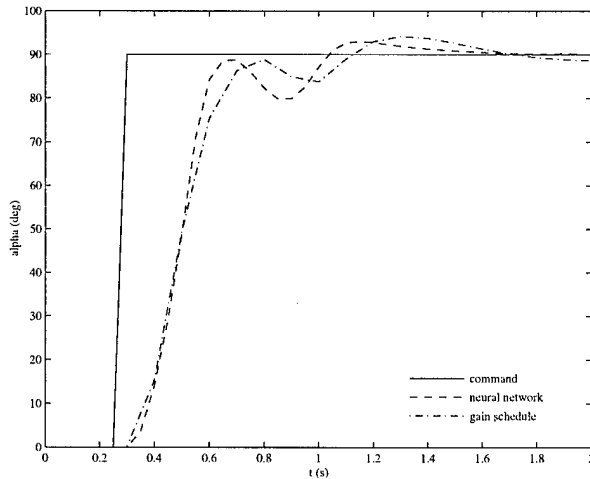
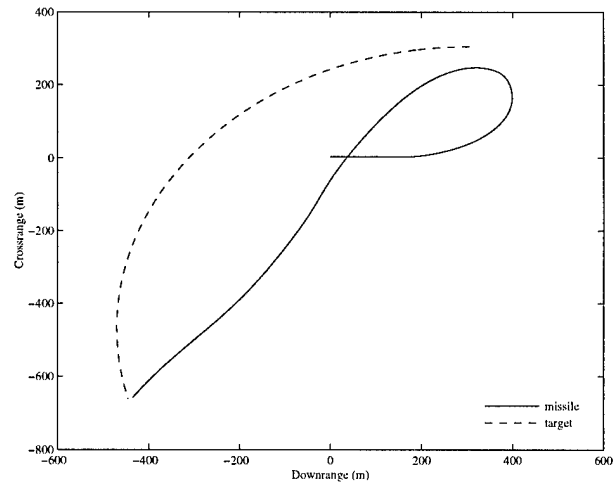Figure 6: Step Response Comparison
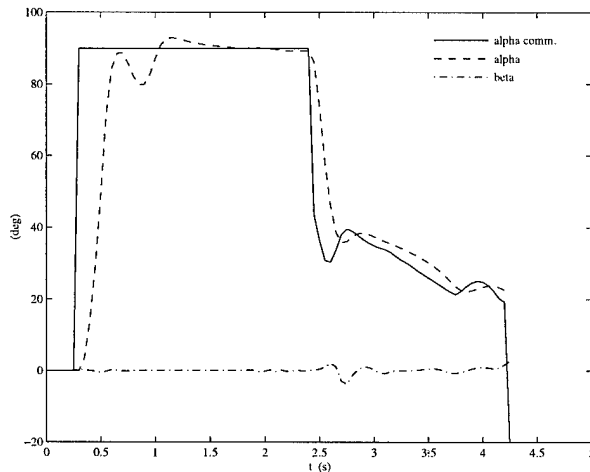

Figure 7: Intercept Trajectories

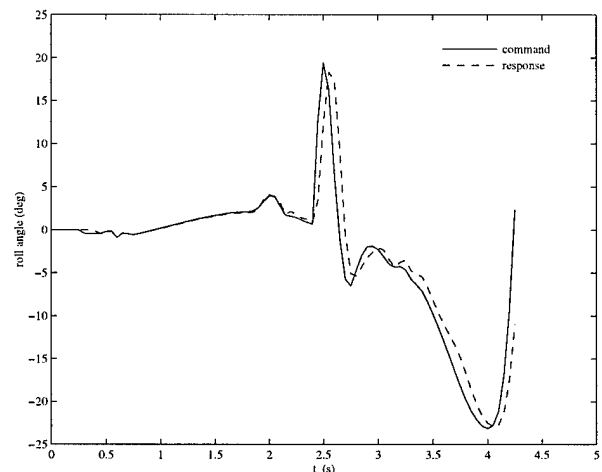Figure 8: Angle of Attack and Sideslip Responses

Figure 9: Roll Angle Response

high-α flight. The autopilot results presented below therefore do not reflect the terminal stage of the intercept, during which control is returned to a gain-scheduled autopilot.

Fig. 6 compares this autopilot with a more conventional gain-scheduled autopilot for a step command of 90 degrees in angle of attack. This plot is representative of 6-DOF simulation results for other maneuvers, which indicate that the two controllers achieve similar performance.

An important engagement geometry for an agile anti-air missile is the so-called "merge" scenario, in which the launch aircraft and the target are passing one another as the missile is fired. In this case the missile initially executes an agile turn at very high angles of attack, and intercept occurs in the rear hemisphere relative to the launch aircraft. Fig. 7 illustrates this scenario, which takes place primarily in the local horizontal plane.

Autopilot angle of attack and sideslip angle tracking responses are depicted in Fig. 8. Sideslip angle is held to less than two degrees throughout the maneuver. The initial angle of attack transient response is a result of both the nonlinear aerodynamics of the missile and the neural network's initial attempts to compensate for these nonlinearities. This will subsequently be discussed in relation to the neural network's effectiveness for error reconstruction.

In Fig. 9, the missile's roll angle response is presented. Implementation concerns which are beyond the scope of this paper dictated that the missile achieve roll control using a redefined roll angle, neither ϕ nor μ. This is merely a detail, however, and does not alter the general control design approach.

Fig's 10 and 11 illustrate the RCS thrust commands. Saturation limits are activated by stressful maneuvering, a phenomenon which has not been explicitly accounted for in the control design process. In the pitch case, such saturation is unavoidable. In yaw, however, high control
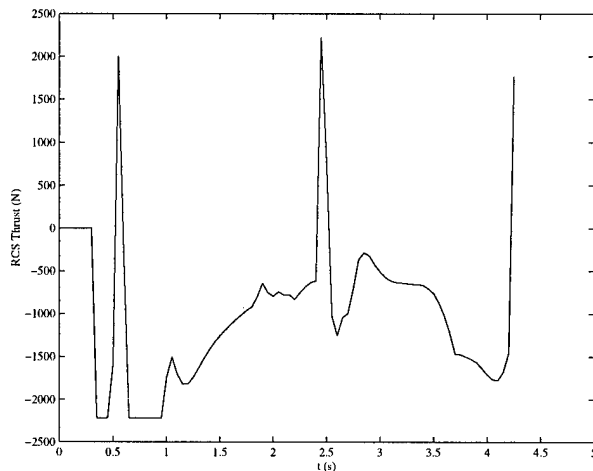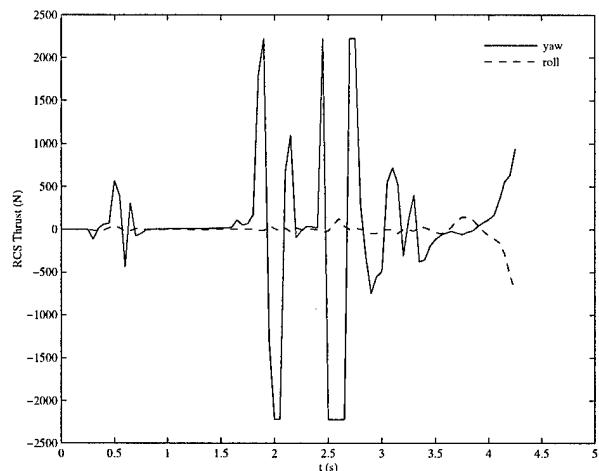
Figure 10: Pitch RCS Thrust
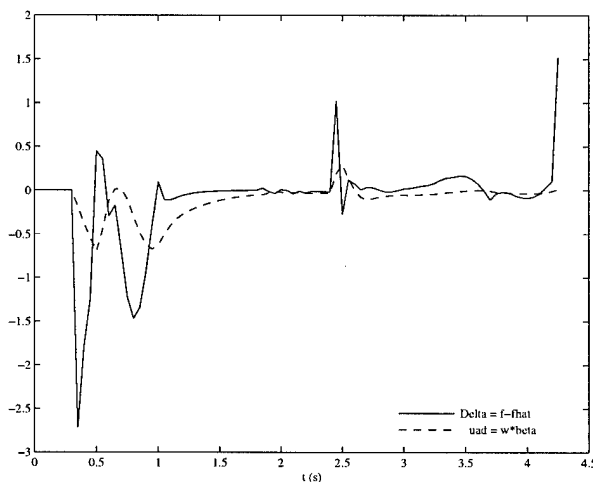

Figure 11: Yaw and Roll RCS Thrust

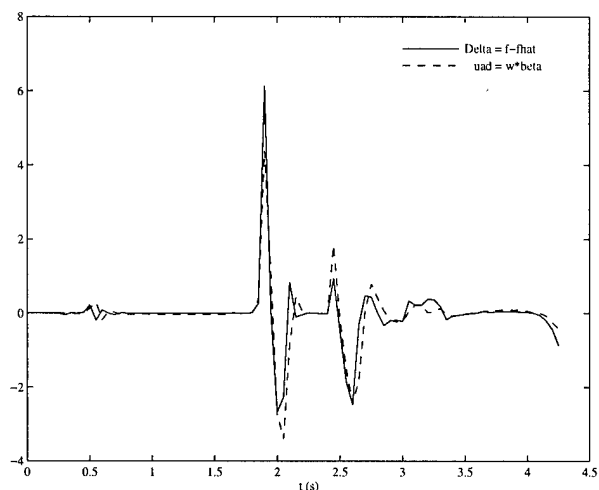Figure 12: Pitch Inversion Error Comparison

Figure 13: Yaw Inversion Error Comparison

activity is a result of the neural network canceling aerodynamic nonlinearities. Reducing the network's adaptation gain (γ) can alleviate this problem, but at the price of increased levels of sideslip.

Finally, Fig's 12 and 13 illustrate the effectiveness of neural networks in inversion error reconstruction. In these figures, the solid lines represent the actual value of the inversion error Δ, which may be computed in the simulation. The dashed lines represent the adaptive pseudo-control component. When the neural network perfectly reconstructs linearization errors, these two quantities have equal values. In Fig. 12, however, there are transient errors for which the network does not compensate completely. Increasing the learning rate (γ) causes the network to respond more aggressively to these transients, but such high gains can lead to undesirable "ringing" as the network attempts to rapidly cancel large transient errors. In the results presented here, adaptation gains were chosen by starting with

small values and increasing the gains until desired performance was achieved. A different neural network structure or choice of basis functions might enable the pitch network to more closely approximate the linearization error.

It is evident from Fig. 13 that the neural network functions as desired for yaw control. This is supported by the fact that sideslip tracking improved as γ was increased in the design process. The roll network performs similarly to the yaw network, and a plot depicting this has been omitted for conciseness.

Conclusions

A nonlinear bank-to-turn missile autopilot based on approximate dynamic inversion has been proposed. With the aid of an adaptive neural network, this autopilot tracks guidance commands in angle of attack and bank angle while holding sideslip angle near zero.

7

A similar development could be carried out for the skid-to-turn case by replacing the bank angle state (μ) with roll angle (ϕ) and maintaining $\phi = \dot{\phi} = 0$.

Nonlinear 6-DOF simulation results indicate that the performance of the neural-adaptive nonlinear autopilot is comparable to that of an existing gain-scheduled autopilot. One advantage of the neural-network-based approach is that it eliminates the time-consuming process of designing a different autopilot at each of numerous flight conditions. Additionally, the use of neural networks enables the nonlinear controller to effectively adapt on-line to uncertain nonlinear aerodynamic phenomena which are difficult to model for purposes of design and simulation.

Neural network based adaptive control of nonlinear systems is a maturing technology area which promises to be applicable to a wide range of systems. The favorable results described here have already led to more advanced research, including the aforementioned robustness results as well as efforts to extend the current approach to accommodate multilayer neural networks. These networks feature improved approximation capabilities over their single-layer counterparts, but at the price of nonlinearity in their adjustable parameters. Each of these advancements represents another step toward the creation of a design technique powerful enough for tomorrow's demanding flight control applications.

References

1. Tahk, M., Briggs, M., and Menon, P.K.A., "Application of Plant Inversion via State Feedback to Missile Autopilot Design," *Proceedings of the 27th Conference on Decision and Control*, Austin, TX, 1986.

2. Innocenti, M., and Thukral, A., "Simultaneous Reaction Jet and Aerodynamic Control of Missile Systems," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, 1993, pp. 347-354.

3. McDowell, D.M., Irwin, G.W., and McConnell, G., "Online Neural Control Applied to a Bank-to-Turn Missile Autopilot," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Baltimore, MD, 1995, pp. 1286-1294.

4. Kim, B.S., and Calise, A.J., "Nonlinear Flight Control Using Neural Networks," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Scottsdale, AZ, 1994, pp. 930-940. (To appear in *AIAA Journal of Guidance, Control and Dynamics*)

5. Leitner, J., Calise, A., and Prasad, J.V.R., "Analysis of Adaptive Neural Networks for Helicopter Flight Controls," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Baltimore, MD, 1995, pp. 871-879.

6. McFarland, M.B., and Calise, A.J., "Neural Networks for Stable Adaptive Control of Air-to-Air Missiles," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Baltimore, MD, 1995.

7. McFarland, M.B., and Calise, A.J., "Neural-Adaptive Nonlinear Autopilot Design for an Agile Anti-Air Missile," AIAA-96-3914, *AIAA Guidance, Navigation and Control Conference*, San Diego, CA, 1996.

8. Krstic, M., Kanellakopoulos, I., and Kokotovic, P., *Nonlinear and Adaptive Control Design*, John Wiley and Sons, New York, 1995.

9. McFarland, M.B., and Calise, A.J., "Robust Adaptive Control of Uncertain Nonlinear Systems Using Neural Networks," manuscript submitted to 1997 American Control Conference.

10. Menon, P.K.A., *et al.*, *Integrated Control and Guidance Systems for High Angle of Attack Missiles*, Final Report, Contract No. F08630-94-C-0015, USAF Wright Laboratory (AFMC), February, 1995.

11. Wise, K.A., and Broy, D.J., "Agile Missile Dynamics and Control," AIAA-96-3912, *AIAA Guidance, Navigation and Control Conference*, San Diego, CA, 1996.

12. Etkin, B., *Dynamics of Atmospheric Flight*, John Wiley & Sons, Inc., New York, 1972, pp. 148-150.

PLEASE CHECK THE APPROPRIATE BLOCK BELOW: $M97-03-2042$

☐ _____ copies are being forwarded. Indicate whether Statement A, B, C, D, E, F, or X applies.

☒ DISTRIBUTION STATEMENT A:
    APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

☐ DISTRIBUTION STATEMENT B:
    DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES
    ONLY; (Indicate Reason and Date). OTHER REQUESTS FOR THIS
    DOCUMENT SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT C:
    DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND
    THEIR CONTRACTORS; (Indicate Reason and Date). OTHER REQUESTS
    FOR THIS DOCUMENT SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT D:
    DISTRIBUTION AUTHORIZED TO DOD AND U.S. DOD CONTRACTORS ONLY;
    (Indicate Reason and Date). OTHER REQUESTS SHALL BE REFERRED TO
    (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT E:
    DISTRIBUTION AUTHORIZED TO DOD COMPONENTS ONLY; (Indicate Reason
    and Date). OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling
    DoD Office).

☐ DISTRIBUTION STATEMENT F:
    FURTHER DISSEMINATION ONLY AS DIRECTED BY (Indicate Controlling DoD
    Office and Date) or HIGHER DOD AUTHORITY.

☐ DISTRIBUTION STATEMENT X:
    DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND PRIVATE
    INDIVIDUALS OR ENTERPRISES ELIGIBLE TO OBTAIN EXPORT-CONTROLLED
    TECHNICAL DATA IN ACCORDANCE WITH DOD DIRECTIVE 5230.25,
    WITHHOLDING OF UNCLASSIFIED TECHNICAL DATA FROM PUBLIC DISCLOSURE,
    6 Nov 1984 (Indicate date of determination). CONTROLLING DOD OFFICE IS
    (Indicate Controlling DoD Office).

☐ This document was previously forwarded to DTIC on _____ (date) and the
    AD number is _____.

☐ In accordance with the provisions of DoD instructions, the document requested is
    not supplied because:

☐ It is TOP SECRET.

☐ It is excepted in accordance with DoD instructions pertaining to communications
    and electronic intelligence.

☐ It is a registered publication.

☐ It is a contract or grant proposal, or an order.

☐ It will be published t a later date. (Enter approximate date, if known.)

☐ Other. (Give Reason.)

Michael B. McFarland
Print or Typed Name

Mich BMcFall 5 DEC 96
Authorized Signature   Date

(404) 816-8074
Telephone Number

⑧⑧ "Nonlinear Adaptive Control of Agile Anti-Air
Missiles Using Neural Networks"
Session 9

DTIC QUALITY INSPECTED 1